LINKÖPINGS UNIVERSITET
Institutionen för datavetenskap
Avdelningen för statistik
Måns Magnusson

2018-06-07
Programmering i R, 7.5 hp
732G33

## Tentamen i Programmering i R, 7.5 hp

---

Skrivtid: 8-12
Hjälpmedel: Inget tryckt material, dock finns referenskort i mappen cheetsheets tillgängligt.
Betygsgränser: Tentamen omfattar totalt 20 poäng. 12 poäng ger Godkänt, 16 poäng ger Väl godkänt.

Tänk på följande:
Skriv dina lösningar i **fullständig och läsbar kod**.
Spara filen med namnet `tentaX.R` där X är ditt SC-nummer (klientnummer).
Det numret kan du se i studentklienten.
Spara filer i mappen `/home/student_tilde/`
Exempel: om du har SC-nummer SC12345 så ska len heta tentaSC12345.R
Notera att du ska lämna in en fill med alla dina lösningar.
Frågor kan ställas till lärare via studentklienten.
När du har loggat ut från datorn är tentan avslutad.
**Kommentera direkt** i R-filen när något behöver förklaras eller diskuteras.
Eventuella grafer behöver **INTE** skickas in för rättning,
det räcker med att **skicka in den kod som producerar figurerna**.

---

**OBS: Glöm inte att spara din fil ofta! Om R krashar kan kod förloras.**

1. Datastrukturer, logik och beräkningar (4p)

   (a) Beräkna $\exp(1 + x) \cdot \cos(\pi/4)$ där $x = \sqrt{2}$ **1p**

   ---

   **Lösningsförslag:**

```
x <- sqrt(2)
exp(1+x)*cos(pi/4)
```

```
[1] 7.90614
```

   ---

   (b) Läs in det interna datamaterialet `ChickWeigh` och räkna ut den genomsnittliga vikten (`weight`). **1p**

   ---

   **Lösningsförslag:**

```
data(ChickWeight)
weight_mean <- mean(ChickWeight$weight)
weight_mean

[1] 121.818
```

---

(c) Beräkna den genomsnittliga vikten (`weight`) efter den kategoriska variabheln kost (`Diet`). Använd funktionen `aggregate()`. **1p**

---

**Lösningsförslag:**

```
weight_by_diet <- aggregate(ChickWeight$weight, by=list(ChickWeight$Diet), FUN=mean)
weight_by_diet

  Group.1       x
1       1 102.645
2       2 122.617
3       3 142.950
4       4 135.263
```

---

(d) Skapa en lista som innehåller dels listelementet med namnet `weight_mean` som innehåller resultatet från (b) ovan och ett listelement du kallar `weight_by_diet` som innehåller en vektor med resultatet från (c). **1p**

---

**Lösningsförslag:**

```
list(weight_mean = weight_mean, weight_by_diet = weight_by_diet[,2])

$weight_mean
[1] 121.818

$weight_by_diet
[1] 102.645 122.617 142.950 135.263
```

---

2. Kontrollstrukturer (4p)

(a) Skriv en `for`-loop som skriver ut alla heltal som är jämt delbara med 11 som finns mellan 1 och 200 med hjälp av en loop och villkorssats.

---

**Lösningsförslag:**

```
for(i in 1:200){
  if(i %% 11 == 0) print(i)
}
```

```
[1] 11
[1] 22
[1] 33
[1] 44
[1] 55
[1] 66
[1] 77
[1] 88
[1] 99
[1] 110
[1] 121
[1] 132
[1] 143
[1] 154
[1] 165
[1] 176
[1] 187
[1] 198
```

---

(b) Skapa med `repeat` och `break` kod som gör följande: **2p**

i. Skriver ut alla jämna tal mellan 9 och 27.

ii. Beräknar och skriver ut resultat i varje steg av den kumulativa summan från 10 till 20.

---

**Lösningsförslag:**

```
i <- 9
repeat{
  if(i %% 2 == 0) print(i)
  if(i == 27) break
  i <- i + 1
}
```

```
[1] 10
[1] 12
```

```
[1] 14
[1] 16
[1] 18
[1] 20
[1] 22
[1] 24
[1] 26
```

```
csum <- 0
i <- 10
repeat{
  csum <- csum + i
  if(i == 20) break
  i <- i + 1
}
csum
```

```
[1] 165
```

---

3. Strängar och datum (4p)

   (a) Läs in paketen `lubridate` och `stringr` i R. **0.5p**

   ---

   **Lösningsförslag:**

```
library(lubridate)
```

```
Warning:  package 'lubridate' was built under R version 3.4.4
```

```
library(stringr)
```

   ---

   (b) Läs in det interna datasetet `lakers`. I detta datmaterial finns variabeln `date`. Beräkna i genomsnitt hur många veckor det senaste datumet är från det tidigaste datumet. **1.5p**

   ---

   **Lösningsförslag:**

```
data(lakers)
dates <- ymd(lakers$date)
min_date <- min(dates)
max_date <- max(dates)
interval(min_date, max_date) / weeks(1)
```

```
[1] 24
```

---

(c) Läs in datasetet `fruit` (ligger i `stringr`-paketet). **0.5p**

---

**Lösningsförslag:**

```
data(fruit)
```

---

(d) Räkna ut med `stringr` det genomsnittliga antalet tecken i `fruit`. **1.5p**

---

**Lösningsförslag:**

```
mean(str_length(fruit))

[1] 8.0875
```

---

4. Funktioner (4p)

Vi ska skapa en funktion kallad `multiplication_table()` som tar ett argument `x` och returnerar en multiplicationstabell upp till och med talet `x`. **4p**

```
multiplication_table(3)

     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    2    4    6
[3,]    3    6    9

multiplication_table(5)

     [,1] [,2] [,3] [,4] [,5]
[1,]    1    2    3    4    5
[2,]    2    4    6    8   10
[3,]    3    6    9   12   15
[4,]    4    8   12   16   20
[5,]    5   10   15   20   25
```

---

**Lösningsförslag:**

```
function(x){
  matrix(1:x, ncol = 1) %*% matrix(1:x, nrow = 1)
}
<bytecode: 0x7fab2082aa38>
```

---

5. Statistik och grafik (4p)

   (a) Läs in datamaterialet `iris` och plocka ut blommorna `setosa` och `versicolor`. **1p**

   .

   ---

   **Lösningsförslag:**

   ```
   iris <- iris[iris$Species != "virginica",]
   pet_setosa <- iris$Petal.Length[iris$Species == "setosa"]
   pet_versicolor <- iris$Petal.Length[iris$Species == "versicolor"]
   ```

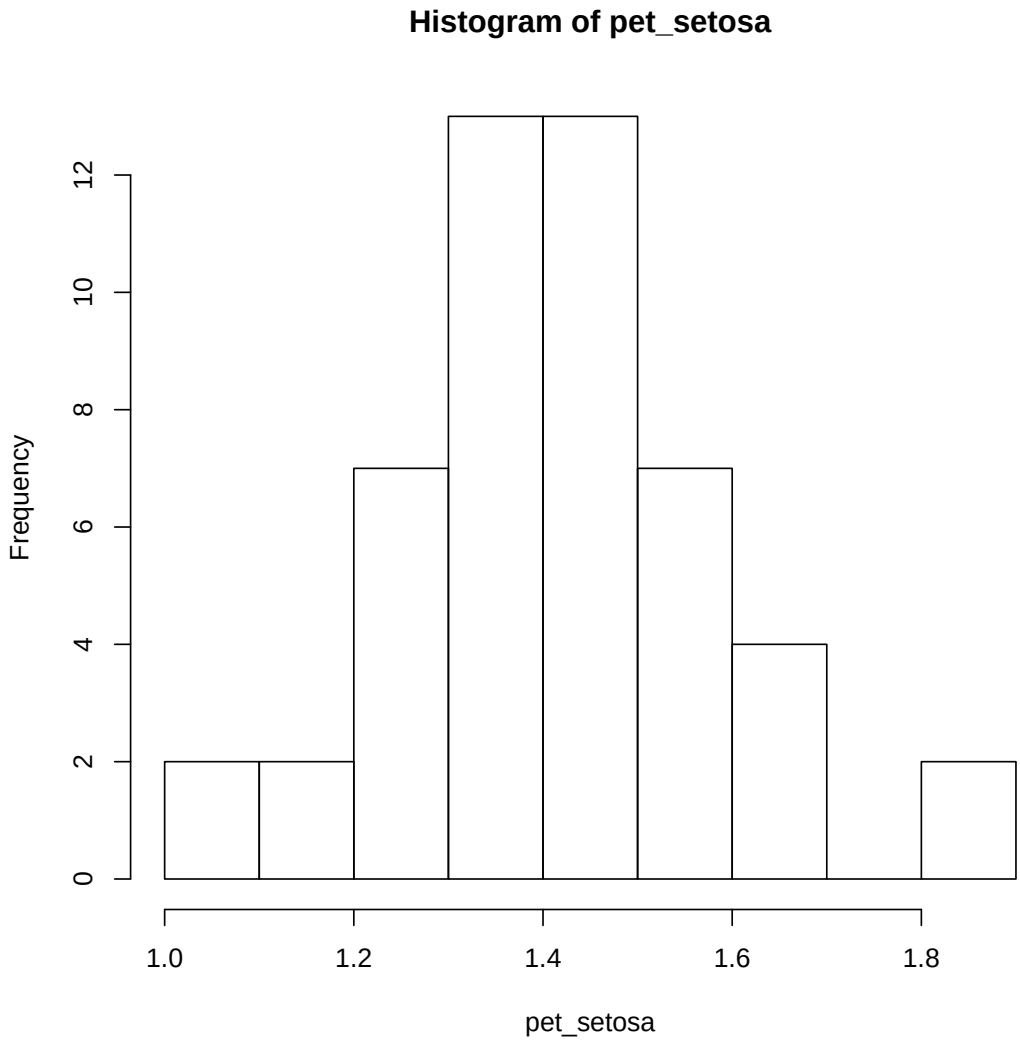
   ---

   (b) Visualisera de två orkidéarternas `petal.length` med två histogram, ett per orkidé. **1.5p**

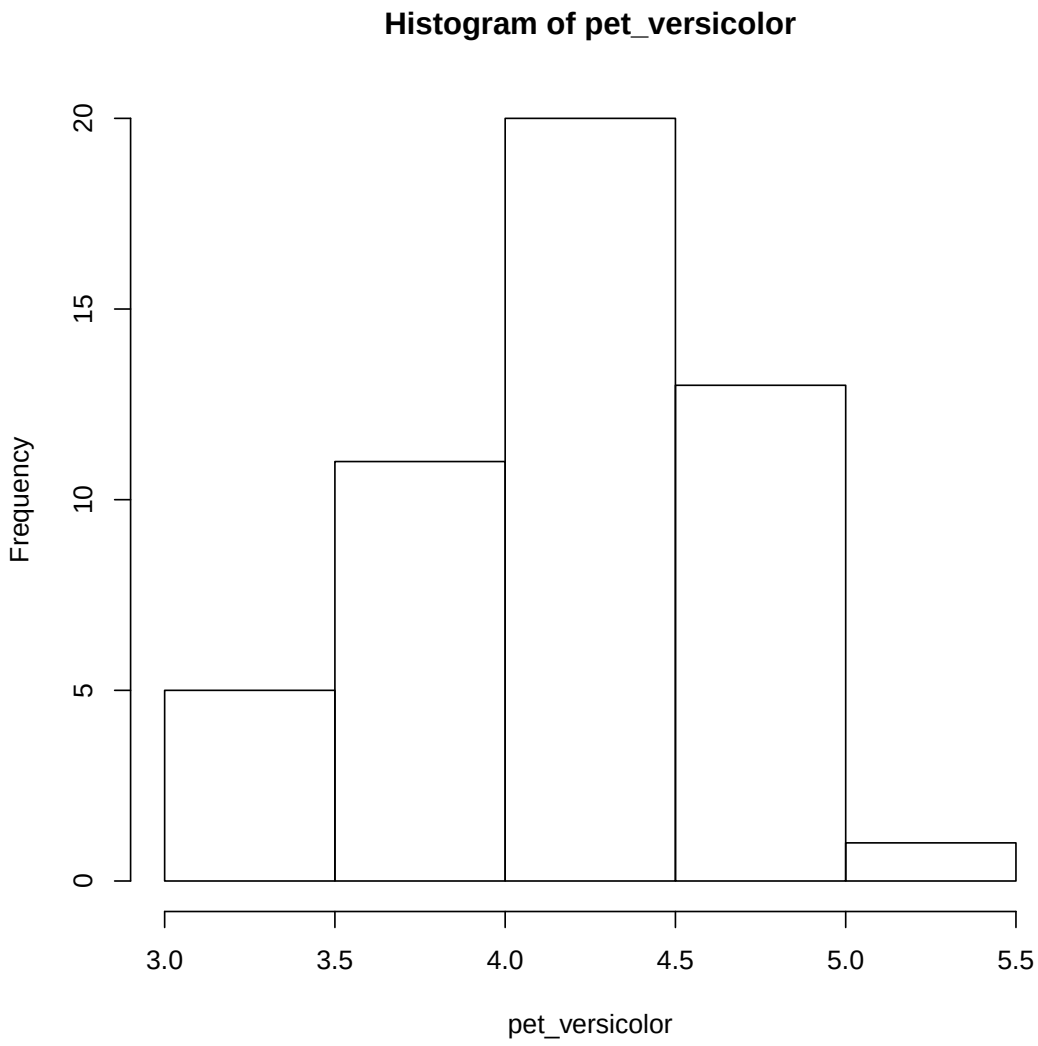
   ---

   **Lösningsförslag:**

   ```
   hist(pet_setosa)
   ```




```
hist(pet_versicolor)
```




---

(c) Gör ett t-test för att testa om det är en signifikant skillnad i `petal.length`. **1.5p**

---

**Lösningsförslag:**

```
t.test(pet_setosa, pet_versicolor)


Welch Two Sample t-test

data:  pet_setosa and pet_versicolor
t = -39.49, df = 62.14, p-value <2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -2.93962 -2.65638
```

```
sample estimates:
mean of x mean of y
    1.462     4.260
```

---

*Lycka till!*